

# Link Scheduling in STDMA Wireless Networks: A Line Graph Approach

N. Praneeth Kumar  
 Goldman Sachs (India) Securities Pvt. Ltd.  
 EGL Business Park, Domlur, Bangalore, India  
 Email: praneeth.kumar@gs.com

Ashutosh Deepak Gore and Abhay Karandikar  
 Department of Electrical Engineering  
 Indian Institute of Technology - Bombay, India  
 Email: {adgore,karandi}@ee.iitb.ac.in

**Abstract**—We consider point to point link scheduling in Spatial Time Division Multiple Access (STDMA) wireless networks under the physical interference model. We propose a novel link scheduling algorithm based on a line graph representation of the network, by embedding the interferences between pairs of nodes into the edge weights of the line graph. Our algorithm achieves lower schedule length and lower run time complexity than existing algorithms.

## I. INTRODUCTION

A prevalent scheme for spatial reuse in wireless networks is Spatial Time Division Multiple Access (STDMA) [1], in which time is divided into fixed-length slots that are organized cyclically. An STDMA link schedule defines the transmission rights for each time slot in such a way that communicating pairs of nodes, i.e., links, assigned to the same time slot do not collide. The interference between the links can be modeled by assuming that transmission on a link is successful if the distance between the nodes is less than some *communication range* and no other node is transmitting within an *interference range* from the receiver of the link. This is called protocol interference model [2]. Under these assumptions, the network can be modeled by a *communication graph* and scheduling algorithms employ edge coloring techniques to minimize the schedule length. Though determining an optimal schedule is known to be NP-complete [3], heuristics have been proposed in [3], [4], [5].

A more realistic model would be to consider a transmission on a link to be successful if the Signal to Interference and Noise Ratio (SINR) at the receiver is greater than some threshold, say  $\gamma_c$ . This model is called physical interference model [2]. Recently, few authors [6], [7], [8], [9] have proposed scheduling algorithms based on this physical interference model, which result in improved network throughput. In [6], the authors propose a polynomial time algorithm which gives a provable performance guarantee within a constant factor compared to the optimal. This algorithm is a greedy heuristic that determines a schedule based on the feasibility of satisfying SINR conditions using a communication graph based representation of the network.

In this paper, we propose a novel scheduling algorithm for STDMA wireless networks under physical interference model. Our approach is based on a line graph [10] representation of the network where the weights of the edges correspond

to interferences between pairs of nodes. Analogous to a line graph, a conflict graph model under physical interference assumptions has been suggested in [11]. However, the authors of [11] do not propose any specific scheduling algorithm and use the weighted conflict graph only to compute bounds on the network throughput. On the other hand, we use a line graph representation of the network under the physical interference model and develop a novel scheduling algorithm with lower time complexity and substantially improved performance in terms of schedule length.

The rest of the paper is organized as follows. In Section II, we explain our system model, formulate the problem and describe the proposed algorithm. In Section III, we prove the correctness of our algorithm and derive its computational complexity. The performance of our algorithm is evaluated in Section IV. We conclude in Section V.

## II. LINE GRAPH BASED LINK SCHEDULING ALGORITHM

Consider an STDMA wireless network  $\Upsilon(\cdot)$  with  $N$  static nodes (wireless routers) in a two-dimensional plane. During a time slot, a node can transmit to exactly one node, receive from exactly one node or remain idle. We assume homogeneous and backlogged nodes. Let:

$$\begin{aligned} P &= \text{transmission power of every node} \\ N_0 &= \text{thermal noise power spectral density} \\ \alpha &= \text{path loss factor} \end{aligned}$$

The received signal power at a distance  $d$  from a transmitter is given by  $\frac{P}{d^\alpha}$ . The STDMA wireless network  $\Upsilon(\cdot)$  is modeled by its communication graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  and  $\mathcal{E}$  denote the sets of vertices and edges respectively, as follows:

- 1) Every node  $i$  in  $\Upsilon(\cdot)$  is represented by a vertex  $v_i \in \mathcal{V}$  in  $\mathcal{G}(\cdot)$ .
- 2) If the Euclidean distance between two distinct nodes  $a$  and  $b$  is no greater than the communication range  $R_c := (\frac{P}{N_0 \gamma_c})^{\frac{1}{\alpha}}$ , then there is a directed edge from  $v_a$  to  $v_b$  in  $\mathcal{G}(\cdot)$ , i.e.,  $(v_a, v_b) \in \mathcal{E}$ .

The set of edges in  $\mathcal{G}(\cdot)$  to be scheduled is determined by a routing algorithm. For simplicity, we only consider exhaustive schedules, i.e., schedules which assign exactly one time slot to every directed edge in  $\mathcal{G}(\cdot)$ .

**Algorithm 1** LineGraphLinkSchedule (LGLS)

---

```

1: Input: Communication graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ ,  $\gamma_c$ ,  $N_0$ ,  $P$ 
2: Output: A coloring  $\mathcal{C}: \mathcal{E} \rightarrow \{1, 2, \dots\}$ 
3:  $\mathcal{V}' \leftarrow \mathcal{E}$  {Phase 1 begins}
4: Construct the directed complete graph  $\mathcal{G}'(\mathcal{V}', \mathcal{E}')$ 
5: for all  $e'_{ij} \in \mathcal{E}'$  do
6:   if edges  $i$  and  $j$  have a common vertex in  $\mathcal{G}(\cdot)$  then
7:      $w_{ij} \leftarrow 1$ 
8:   else
9:      $w_{ij} \leftarrow \gamma_c \frac{d(t_j, r_j)^\alpha}{d(t_i, r_j)^\alpha}$ 
10:  end if
11: end for {Phase 1 ends}
12: for all  $e'_{ij} \in \mathcal{E}'$  do {Phase 2 begins}
13:    $w'(e'_{ij}) \leftarrow \max\{0, 1 - w_{ij}\}$ 
14: end for {Phase 2 ends}
15: for all  $v'_j \in \mathcal{V}'$  do {Phase 3 begins}
16:    $\mathcal{N}(v'_j) \leftarrow \frac{N_0 \gamma_c}{P} d(t_j, r_j)^\alpha$ 
17: end for {Phase 3 ends}
18:  $\mathcal{V}'_{uc} \leftarrow \mathcal{V}'$  {Phase 4 begins}
19:  $p \leftarrow 0$ 
20: while  $\mathcal{V}'_{uc} \neq \emptyset$  do
21:    $p \leftarrow p + 1$ 
22:   choose  $v' \in \mathcal{V}'_{uc}$  randomly
23:    $\mathcal{C}(v') \leftarrow p$ 
24:    $\mathcal{V}'_{uc} \leftarrow \mathcal{V}'_{uc} \setminus \{v'\}$ 
25:    $\mathcal{V}'_p \leftarrow \{v'\}$ 
26:    $\psi \leftarrow 1$ 
27:   while  $\psi = 1$  and  $\mathcal{V}'_{uc} \neq \emptyset$  do
28:      $u' \leftarrow \arg \max_{y' \in \mathcal{V}'_{uc}} \sum_{x' \in \mathcal{V}'_p} w'(e'_{x'y'}) + w'(e'_{y'x'})$ 
29:     for all  $v'_c \in \mathcal{V}'_p$  do
30:       if  $\sum_{v'_1 \in \mathcal{V}'_p \setminus \{v'_c\} \cup \{u'\}} w'(e'_{v'_1 v'_c}) \leq |\mathcal{V}'_p| + \mathcal{N}(v'_c) - 1$ 
31:         then
32:            $\psi \leftarrow 0$ 
33:         end if
34:       end for
35:       if  $\psi = 1$  and  $\sum_{v'_1 \in \mathcal{V}'_p} w'(e'_{v'_1 u'}) > |\mathcal{V}'_p| + \mathcal{N}(u') - 1$ 
36:         then
37:            $\mathcal{C}(u') \leftarrow p$ 
38:            $\mathcal{V}'_p \leftarrow \mathcal{V}'_p \cup \{u'\}$ 
39:            $\mathcal{V}'_{uc} \leftarrow \mathcal{V}'_{uc} \setminus \{u'\}$ 
40:         else
41:            $\psi \leftarrow 0$ 
42:         end if
43:       end while
44:     end while {Phase 4 ends}

```

---

We now motivate our Line Graph based Link Scheduling (LGLS) algorithm, and provide the pseudocode in Algorithm 1. In Phase 1, we first construct a directed complete line graph [10]  $\mathcal{G}'(\mathcal{V}', \mathcal{E}')$  which has the edges of  $\mathcal{G}(\cdot)$  as its vertices, i.e.,  $\mathcal{V}' = \mathcal{E}$ . Let the edges of  $\mathcal{G}(\cdot)$  and the corresponding vertices of  $\mathcal{G}'(\cdot)$  be labeled  $1, 2, \dots, e$ . Let  $t_i$  and  $r_i$  denote the transmitter and receiver respectively of edge  $i$  in  $\mathcal{G}(\cdot)$ . Let  $d(t_i, r_j)$  denote the Euclidean distance between  $t_i$  and  $r_j$ .

For any two edges  $i$  and  $j$  in graph  $\mathcal{G}(\cdot)$ , the weight function  $w_{ij}$  is defined as:

$$w_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ have a common vertex,} \\ \gamma_c \frac{d(t_j, r_j)^\alpha}{d(t_i, r_j)^\alpha} & \text{otherwise.} \end{cases}$$

This weight function  $w_{ij}$  indicates the interference energy at  $r_j$  due to transmission from  $t_i$  to  $r_i$  scaled with respect to the signal energy of  $t_j$  at  $r_j$ .

In Phase 2, we compute the co-schedulability weight function  $w'(\cdot)$ . For any two edges  $i$  and  $j$  in  $\mathcal{G}(\cdot)$ , the weight of the edge  $e'_{ij}$  in  $\mathcal{G}'(\cdot)$  is given by  $w'(e'_{ij}) = \max\{0, 1 - w_{ij}\}$ . Since  $w_{ij}$  and  $w_{ji}$  represent the interference between links  $i$  and  $j$  in  $\Upsilon(\cdot)$ ,  $w'(e'_{ij})$  and  $w'(e'_{ji})$  intuitively represent the co-schedulability of vertices  $i$  and  $j$  in  $\mathcal{G}'(\cdot)$ . For example, if  $w_{ij}$  is greater than or equal to 1, it means that the interference between links  $i$  and  $j$  in  $\Upsilon(\cdot)$  is very high and these links cannot be scheduled simultaneously. This will result in  $w'(e'_{ij})$  being equal to 0 indicating that the vertices  $i$  and  $j$  in  $\mathcal{G}'(\cdot)$  are not co-schedulable. On the other hand, if  $w_{ij}$  is almost 0,  $w'(e'_{ij})$  will be almost 1 indicating that the vertices  $i$  and  $j$  in  $\mathcal{G}'(\cdot)$  are co-schedulable. In Phase 3, we determine the normalized noise power at the receiver of each vertex of  $\mathcal{G}'(\cdot)$ .

Our objective is to color the vertices of  $\mathcal{G}'(\cdot)$  (equivalently, edges of  $\mathcal{G}(\cdot)$ ) using minimum number of colors under the physical interference model, i.e., subject to the condition that the SINR at every receiver of each link in  $\Upsilon(\cdot)$  is greater than the communication threshold  $\gamma_c$ . Equivalently, for any  $\mathcal{V}'_{cc} \subseteq \mathcal{V}'$ , the coloring of all nodes  $v'_i \in \mathcal{V}'_{cc}$  with the same color is defined to be *feasible* if

$$\frac{\frac{P}{d(t_{v'_i}, r_{v'_i})^\alpha}}{N_0 + \sum_{v'_j \in \mathcal{V}'_{cc} \setminus \{v'_i\}} \frac{P}{d(t_{v'_j}, r_{v'_j})^\alpha}} \geq \gamma_c \quad \forall v'_i \in \mathcal{V}'_{cc} \quad (1)$$

In  $\mathcal{G}'(\cdot)$ , this condition translates to the sum of weights of edges incoming to a vertex from all the co-colored vertices being greater than the sum of the number of *remaining* co-colored vertices and the normalized noise power minus a constant factor (unity).

The actual coloring of vertices of  $\mathcal{G}'(\cdot)$ , i.e., edges of  $\mathcal{G}(\cdot)$ , occurs in Phase 4. Let  $\mathcal{V}'_{uc}$  at any time denote the set of uncolored vertices of  $\mathcal{G}'(\cdot)$  till that time. Initially,  $\mathcal{V}'_{uc}$  includes all the vertices of  $\mathcal{G}'(\cdot)$ . First we choose a vertex randomly from  $\mathcal{V}'_{uc}$ . This is assigned a new color, let it be  $p$ . Then we choose that vertex from  $\mathcal{V}'_{uc}$  which maximizes the sum of weights of all the edges between that vertex and the vertices colored by  $p$ . Now for each vertex colored with  $p$ , we check if the sum of weights of all the incoming edges is greater than the sum of the number of vertices colored with  $p$  and the normalized noise power minus a constant factor (unity). If it is satisfied, the vertex is colored with  $p$ . If not, it is colored with a new color. The algorithm exits when all the vertices are colored.

### III. ANALYSIS OF LGLS ALGORITHM

In this section, we prove the correctness of the LGLS algorithm and derive its run time (computational) complexity.

We follow the notation of Algorithm 1.

**Theorem 1:** For any  $\mathcal{V}'_{cc} \subseteq \mathcal{V}'$ , if  $\sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} w'(e'_{v'_2 v'_1}) > |\mathcal{V}'_{cc}| + \mathcal{N}(v'_1) - 2$   $\forall v'_1 \in \mathcal{V}'_{cc}$ , then the coloring of all the vertices of  $\mathcal{V}'_{cc}$  with the same color is feasible.

**Proof:**  $w'(e'_{v'_2 v'_1}) \in [0, 1]$ . Suppose  $w'(e'_{v'_3 v'_1}) = 0$  for some  $v'_1, v'_3 \in \mathcal{V}'_{cc}$ ,  $v'_1 \neq v'_3$ , then  $\sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} w'(e'_{v'_2 v'_1}) = \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1, v'_3\}} w'(e'_{v'_2 v'_1}) \leq \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1, v'_3\}} 1 = |\mathcal{V}'_{cc} \setminus \{v'_1, v'_3\}| = |\mathcal{V}'_{cc}| - 2$ , which contradicts the hypothesis. So, an edge connecting any two vertices in  $\mathcal{V}'_{cc}$  has positive weight. But,  $w'(e'_{v'_2 v'_1}) = 0$  or  $1 - w_{v'_2 v'_1}$ . So,  $w'(e'_{v'_2 v'_1}) = 1 - w_{v'_2 v'_1} \forall v'_1, v'_2 \in \mathcal{V}'_{cc}$ ,  $v'_1 \neq v'_2$  and  $w_{v'_2 v'_1} < 1 \forall v'_1, v'_2 \in \mathcal{V}'_{cc}$ ,  $v'_1 \neq v'_2$ . If two vertices  $v'_1, v'_2 \in \mathcal{V}'_{cc}$  have a common vertex in  $\mathcal{G}(\cdot)$ , then  $w_{v'_2 v'_1} = 1$ , which is a contradiction. So no two vertices in  $\mathcal{V}'_{cc}$  have a common vertex in  $\mathcal{G}(\cdot)$ . From the hypothesis,

$$\begin{aligned} \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} w'(e'_{v'_2 v'_1}) &> |\mathcal{V}'_{cc}| + \mathcal{N}(v'_1) - 2 \quad \forall v'_1 \in \mathcal{V}'_{cc} \\ \Leftrightarrow \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} (1 - w_{v'_2 v'_1}) &> |\mathcal{V}'_{cc}| + \mathcal{N}(v'_1) - 2 \\ \Leftrightarrow |\mathcal{V}'_{cc} \setminus \{v'_1\}| - \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} w_{v'_2 v'_1} &> |\mathcal{V}'_{cc}| + \mathcal{N}(v'_1) - 2 \\ \Leftrightarrow |\mathcal{V}'_{cc}| - 1 - \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} w_{v'_2 v'_1} &> |\mathcal{V}'_{cc}| + \mathcal{N}(v'_1) - 2 \\ \Leftrightarrow \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} w_{v'_2 v'_1} + \mathcal{N}(v'_1) &< 1 \\ \Leftrightarrow \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} \gamma_c \frac{d(t_{v'_1}, r_{v'_1})^\alpha}{d(t_{v'_2}, r_{v'_1})^\alpha} + \frac{N_0 \gamma_c}{P} d(t_{v'_1}, r_{v'_1})^\alpha &< 1 \\ \Leftrightarrow \frac{\frac{P}{d(t_{v'_1}, r_{v'_1})^\alpha}}{N_0 + \sum_{v'_2 \in \mathcal{V}'_{cc} \setminus \{v'_1\}} \frac{P}{d(t_{v'_2}, r_{v'_1})^\alpha}} &> \gamma_c \quad \forall v'_1 \in \mathcal{V}'_{cc}. \end{aligned}$$

Therefore, the SINR threshold condition (1) is satisfied at all the receivers of all the vertices of  $\mathcal{V}'_{cc}$ .  $\blacksquare$

With respect to (w.r.t.) the communication graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ , let:

$$\begin{aligned} e &= \text{number of edges} \\ v &= \text{number of vertices} \end{aligned}$$

**Theorem 2:** The run time complexity of LGLS algorithm under uniform load conditions is  $O(e^2)$ .

**Proof:**  $|\mathcal{V}'| = |\mathcal{E}|$ . Since  $\mathcal{G}'(\cdot)$  is a directed complete graph,  $|\mathcal{E}'| = |\mathcal{V}'|^2 = e^2$ . Since the computation of the function  $w_{ij}$  for a given  $i$  and  $j$  takes unit time, the computation of  $w_{ij}$  for all edges  $i$  and  $j$  of  $\mathcal{G}(\cdot)$  takes  $O(e^2)$  time. Hence, the time complexity of Phase 1 is  $O(e^2)$ . Similarly, the time complexity of Phase 2 is also  $O(e^2)$ . The time complexity of Phase 3 is  $O(e)$ . In  $\mathcal{G}'(\cdot)$ , let  $C$  denote the total number of colors used to color all vertices, and let  $N_i$  denote the number of vertices assigned color  $i$ . Since  $C$  can never exceed the number of vertices in  $\mathcal{G}'(\cdot)$  (the number of edges in  $\mathcal{G}(\cdot)$ ),  $C$  is  $O(e)$ . The time required by Lines 21-26 is  $O(1)$ , let it be  $k_1$ , where  $k_1$  is a constant. By using a careful implementation of storing  $\sum_{x' \in \mathcal{V}'_p} w'(e'_{x' y'})$  and  $\sum_{x' \in \mathcal{V}'_p} w'(e'_{y' x'}) \forall y' \in \mathcal{V}'_{uc}$ , the time required by Line 28 will be  $O(|\mathcal{V}'_{uc}|)$ . Let it be equal to  $a_1 |\mathcal{V}'_{uc}|$ , where  $a_1$  is a constant. With a careful implementation of storing  $\sum_{x' \in \mathcal{V}'_p \setminus \{y'\} \cup \{u'\}} w'(e'_{x' y'}) \forall y' \in \mathcal{V}'_p$ , Lines 30-32 take  $O(1)$  time. Hence, Lines 29-33 take time  $O(|\mathcal{V}'_p|)$ , let it be  $a_2 |\mathcal{V}'_p|$ , where  $a_2$  is a constant. Lines 34-40 take  $O(|\mathcal{V}'_p|)$  time, let it be  $a_3 |\mathcal{V}'_p|$ , where  $a_3$  is a constant. Thus the total run time of Phase 4 is,

$$\tau = \sum_{i=1}^C (k_1 + \sum_{m=1}^{N_i} (a_1 |\mathcal{V}'_{uc}| + (a_2 + a_3) |\mathcal{V}'_p|)).$$

$$\begin{aligned} \mathcal{V}'_{uc}, \mathcal{V}'_p \subseteq \mathcal{V}' &\Rightarrow |\mathcal{V}'_{uc}|, |\mathcal{V}'_p| \leq |\mathcal{V}'| = e. \\ \Rightarrow \tau &\leq \sum_{i=1}^C \left( k_1 + \sum_{m=1}^{N_i} (a_1 + a_2 + a_3) e \right) \\ &= \sum_{i=1}^C k_1 + (a_1 + a_2 + a_3) e \sum_{i=1}^C \sum_{m=1}^{N_i} \\ &= k_1 C + (a_1 + a_2 + a_3) e (e) \\ &= O(e^2) \end{aligned}$$

Hence, the total time complexity of Algorithm 1 is  $O(e^2)$ .  $\blacksquare$

We compare our work with that of [6], since it is the only work in STDMA link scheduling whose system model is closest to our system model.

Under uniform load on all edges, the time complexity of GreedyPhysical (GP) algorithm of [6] is  $O(ve^2)$ , whereas the time complexity of the proposed LGLS algorithm is  $O(e^2)$ . Thus, under uniform load conditions, the time complexity of LGLS is much lower than that of GP.

Note that our algorithm can be easily extended to incorporate non-uniform load conditions by assigning integer weights to edges in  $\mathcal{G}(\cdot)$  and then considering each edge split into those many edges. Similar to [6], we assume that the ratio of the maximum load at any node to the minimum load at any node is upper bounded by a constant.

**Corollary 1:** The run time complexity of LGLS algorithm under non-uniform load conditions is  $O(v^2 e^2)$ .

**Proof:** Let  $l_{max}$  and  $l_{min}$  denote the maximum load and minimum load offered by a node in  $\mathcal{G}(\cdot)$ . We assume that  $\frac{l_{max}}{l_{min}}$  is upper bounded by  $k$ , a constant. We normalize the load at each node w.r.t.  $l_{min}$ . So, the load at each node is  $i$ , where  $1 \leq i \leq k$ . Thus, the weight on each edge in  $\mathcal{G}(\cdot)$  is upper bounded by  $\sum_{v \in \mathcal{V}} i \leq vk$ , which is  $O(v)$ . Hence, the total number of edges after replacing an edge by the number of edges equal to its weight is  $O(ve)$ . Therefore, along similar lines as the proof of Theorem 2, the total time complexity is  $O(v^2 e^2)$ .  $\blacksquare$

Thus, under non-uniform load on all edges, the time complexity of LGLS is the same as the time complexity of GP (see [6], Section 4).

#### IV. PERFORMANCE RESULTS

In our simulations,  $N$  nodes are scattered uniformly in a square by choosing the  $X$  and  $Y$  coordinates uniformly from  $[0, L]$ , where  $L = 3000$  m. The system parameters are  $\alpha = 4.5$ ,  $\gamma_c = 7$  dB,  $P = 1000$  mW and  $N_0 = -96$  dBm, which are chosen to match IEEE 802.11b operation values [12]. The LGLS algorithm is compared with the GP algorithm [6] with a weight of 1 on all the edges. We vary  $N$  from 25 to 250 in steps of 25. The schedule length is averaged over 200 random graphs for each value of  $N$ . Fig. 1 plots the average schedule length versus the number of nodes. We observe that LGLS achieves 50-93% lower schedule length than GP.

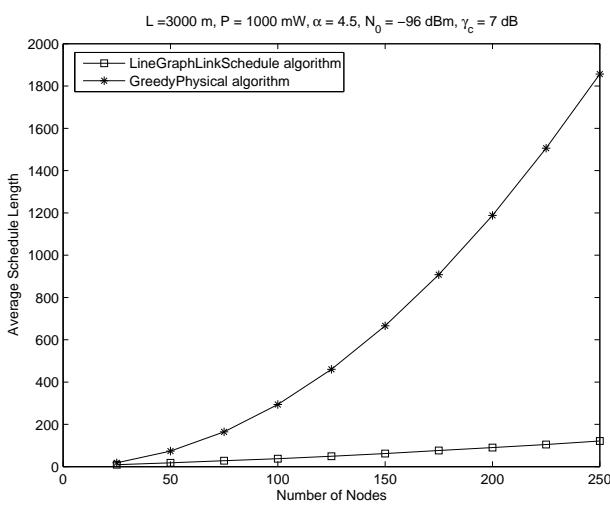


Fig. 1. Schedule length vs. number of nodes for LGSL and GP algorithms.

## V. CONCLUSIONS

In this paper, we have proposed a novel scheduling algorithm based on a line graph representation of STDMA network under the physical interference model. Our results demonstrate that the schedule length for the proposed algorithm is substantially lower than that of the GreedyPhysical algorithm. This is due to the fact that we have embedded SINR feasibility conditions into the edge weights of the line graph, and consequently determined a conflict-free schedule.

## REFERENCES

- [1] R. Nelson and L. Kleinrock, "Spatial TDMA: A Collision-Free Multihop Channel Access Protocol," *IEEE Trans. Commun.*, vol. 33, no. 9, pp. 934–944, Sep. 1985.
- [2] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Inf. Theory*, vol. 46, pp. 388–404, Mar. 2000.
- [3] S. Ramanathan and E. L. Lloyd, "Scheduling Algorithms for Multihop Radio Networks," *IEEE/ACM Trans. Netw.*, vol. 1, no. 2, pp. 166–177, Apr. 1993.
- [4] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient Interference-Aware TDMA Link Scheduling for Static Wireless Networks," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006, pp. 262–273.
- [5] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks," in *Proc. ACM MobiCom*, Cologne, Germany, Aug.-Sep. 2005, pp. 58–72.
- [6] G. Brar, D. M. Blough, and P. Santi, "Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006, pp. 2–13.
- [7] T. Moscibroda and R. Wattenhofer, "The Complexity of Connectivity in Wireless Networks," in *Proc. IEEE Infocom*, Barcelona, Spain, Apr. 2006.
- [8] A. D. Gore, S. Jagabathula, and A. Karandikar, "On High Spatial Reuse Link Scheduling in STDMA Wireless Ad Hoc Networks," accepted for publication in IEEE Globecom 2007.
- [9] A. D. Gore and A. Karandikar, "On High Spatial Reuse Broadcast Scheduling in STDMA Wireless Ad Hoc Networks," in *Proc. National Conference on Communications*, Kanpur, India, Jan. 2007, pp. 74–78.
- [10] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, 2000.
- [11] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Wireless Networks*, vol. 11, no. 4, pp. 471–487, Jul. 2005.
- [12] M. V. Clark, K. K. Leung, B. McNair, and Z. Kostic, "Outdoor IEEE 802.11 Cellular Networks: Radio Link Performance," in *Proc. IEEE ICC*, Apr.-May 2002, pp. 512–516.